Making Non-Centralized a Model Predictive Control Scheme by Using Distributed Smith Dynamics ⋆

J. Barreiro-Gomez [*,**] **G. Obando** [**] **C. Ocampo-Martinez** [*]
N. Quijano [**]

[*] *Automatic Control Department, Universitat Politècnica de Catalunya, Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas, 4-6, 08028 Barcelona, Spain (e-mail: {jbarreiro, cocampo}@ iri.upc.edu)*
[**] *Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Carrera 1 No 18A-10, Bogotá, Colombia (e-mail: {j.barreiro135, ge-oband, nquijano}@ uniandes.edu.co)*

Abstract: This paper proposes a non–centralized Model Predictive Control (MPC) scheme for a system comprised by several sub-systems. Operational constraints for each sub–system are considered as well as a single coupled constraint on the control inputs that models a limitation of the *resource* supplied by the controller. If the underlying optimization problem is of large-scale nature, traditional MPC suffers from computational burden issues. A cause of this problem is the requirement of having centralized information to guarantee that the computed control actions satisfy the coupled constraint. In this work, a traditional MPC is made non–centralized by means of a strategy based on distributed population dynamics. The proposed methodology divides the problem into several local MPC controllers that coordinate their decisions by using a communication network without the need of a centralized scheme. It is proved that this methodology provides an optimal solution that satisfies both the operational constraints of each sub–system, and the coupled constraint of the control signals. Finally, the proposed method is compared with a traditional centralized MPC in an industrial problem that involves several continuously stirred tank reactors.

Keywords: Predictive control, population dynamics, non-centralized control, plug and play.

1. INTRODUCTION

Model Predictive Control (MPC) involves a prediction model and an optimization process, which allows to design closed–loop systems with a desired stationary state value and a desired performance. However, when coping with the control of a large–scale system, there would be a large number of decision variables and constraints that make difficult to guarantee the computation of the optimal control action within the established sampling time (Conte et al. (2012)). Under this scenario, a possible solution is to divide the original problem into smaller and computationally lighter sub-problems, which could be separately solved by using local hardware.

For a large–scale system composed by several sub-systems, it is common to have a constraint related to the total energy resource available for the control actions, i.e., in real applications, the total energy demanded by the controllers has an upper bound since the employed resources (e.g., inflows, voltages, forces, etc.) are limited. Traditional MPC schemes are capable to overcome this problem by adding that consideration into the set of constraints. Nonetheless, this solution requires to have information about the whole system, which implies a centralized control structure that commonly suffers from computational burden issues. Therefore, non–centralized control methods might be an alternative. The problem of obtaining non–centralized control formulations has become an appealing research topic (Christofides et al. (2013)). Some authors propose the decomposition of the overall control problem into smaller decoupled problems and the coordination of those individual components in a centralized way (Spudic and Baotić (2013)). In Wakasa et al. (2008), a dual decomposition is presented to design decentralized MPC controller considering coupled constraints for system outputs. Additionally, a partitioning method for large–scale complex systems based on graph theory is proposed in Ocampo-Martinez et al. (2011). On the other hand, game theory and population dynamics have been recently used as a tool to obtain distributed algorithms for controlling engineering problems (Marden and Shamma (2015)). For instance, different applications in water management, smart lighting, dispatch of electric generators, smart buildings, and communication

⋆ This work has been partially supported by the projects ECOCIS (Ref. DPI2013-48243-C2-1-R), EFFINET (Ref. FP7-ICT-2011-8-31855), and "Drenaje urbano y cambio climático: Hacia los sistemas de alcantarillados del futuro. Fase II. COLCIENCIAS 633/2013".
J. Barreiro-Gomez and G. Obando are partially supported by Colciencias-Colfuturo. J. Barreiro-Gomez is also partially supported by Agència de Gestió d'Ajust Universitaris i de Recerca AGAUR.

systems are presented in Pantoja and Quijano (2011), Obando et al. (2014), and Tembine et al. (2010). Most of the aforementioned papers deal with the problem of dynamic resource allocation.

The contribution of this paper is to state a non-centralized scheme for a traditional constrained MPC to manage medium/large–scale systems comprised by several sub–systems. The formulation considers operational and physical constraints for each sub-system, and a single coupled constraint associated to a limited resource. The latter constraint involves the interaction of all the control signals of the whole system. First, it is proposed to design a local MPC controller per sub–system that is in charge of managing the desired local variables. Then, all the controller outputs are optimally coordinated without the need of a centralized configuration. This coordination process is performed by means of a population dynamics approach, specifically by using the distributed Smith dynamics (Barreiro-Gomez et al. (2014)) [1].

The remainder of this paper is organized as follows. In Section 2, population dynamics background, and the Smith dynamics are presented. Section 3 states the control goals and the constraints of the problem under consideration. In Section 4, the non-centralized MPC topology and the proposed methodology are explained. In Section 5, a case study is presented and simulation results of traditional centralized and non-centralized MPC with distributed population dynamics are compared. Finally, conclusions are drawn in Section 6.

2. POPULATION DYNAMICS

Consider a population composed by a large and finite number of rational agents [2] involved in a strategic game. During this game, each agent chooses a strategy from the set of the M available strategies, which is denoted by $S = \{1, ..., M\}$, trying to improve its benefits. Let $p_i \in \mathbb{R}_{\geq 0}$ be the portion of agents choosing the strategy $i \in S$. Thus, the vector $p = [p_1, \ldots, p_M]^\top$ corresponds to the distribution of agents among the available strategies, i.e., p describes the population state. The set of possible states in the population is given by a simplex denoted by $\Delta = \{p \in \mathbb{R}^M_{\geq 0} : p^\top \mathbb{1}_M = K\}$, where $\mathbb{1}_M$ is a column vector with unitary entries and cardinality $|\mathbb{1}_M| = M$. The interior of the set of population states is denoted by $\text{int}\Delta = \left\{p \in \mathbb{R}^M_{>0} : p^\top \mathbb{1}_M = K\right\}$. Moreover, K is a constant value associated to the total population mass. The fitness function $F_i : \Delta \mapsto \mathbb{R}$ takes a state in the population and returns a real value corresponding to the payoff that the portion of agents p_i receives for playing the strategy $i \in S$. Similarly, the fitness function $F : \Delta \mapsto \mathbb{R}^M$ is the column vector of all fitness functions.

In order to propose the non-centralized MPC scheme, this paper focuses only on full potential games, i.e., there exists a continuously differentiable function $f(p)$ (known as potential function) satisfying $F(p) = \nabla f(p)$, for all $p \in \mathbb{R}^M$. The reason to concentrate this work on this type of games is that in a full potential game, the Nash equilibrium of $F(p)$, denoted by $p^* \in \Delta$, solves the following optimization problem:

$$\max \ f(p) \tag{1a}$$
$$\text{subject to } \ p^\top \mathbb{1}_M = K \tag{1b}$$
$$p_i \geq 0, \text{ for all } \ i \in S, \tag{1c}$$

where $f(p)$ is a strictly concave function.

Theorem 1. If f is strictly concave on Δ, then the Nash equilibrium of the corresponding full potential game is the unique maximizer of f on Δ (set of possible population states) [3].

Distributed Smith Dynamics (DSD): The traditional Smith dynamics equation is one of the six fundamental population dynamics (Sandholm (2010)) and requires full information (i.e., each portion of agents playing a strategy requires information about all agents playing a different strategy to evolve). However, the distributed Smith dynamics are deduced in Barreiro-Gomez et al. (2014) from local revision protocols that only need partial information. Since local revision protocols rely on non-well mixed population, it is assumed that the population interaction is described by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V}$ is the set of nodes, which represents the possible strategies in the game. Besides, $\mathcal{E} \subset \{(i, j) : i, j \in \mathcal{V}\}$ is the set of links representing possible interaction among strategies, and $\mathcal{A}$ is the adjacency matrix whose element $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. DSD are given by

$$\dot{p}_i = \sum_{j \in \mathcal{N}_i} p_j [F_i - F_j]_+ - p_i \sum_{j \in \mathcal{N}_i} [F_j - F_i]_+, \tag{2}$$

where $[\cdot]_+ = \max(0, \cdot)$, and $\mathcal{N}_i = \{j : (i, j) \in \mathcal{E}\}$ is the set of neighbors of the node $i \in \mathcal{V}$. Notice that $i \notin \mathcal{N}_i$.

Proposition 2. If $p(0) \in \Delta$, the simplex Δ is an invariant set under the DSD in (2) [4].

Proposition 3. Let $F(p)$ be a full potential game with strictly concave potential function $f(p)$. Let $\dot{p}$ be the DSD in (2), and let $p^* = \arg\max_{p \in \Delta} f(p)$. If $\mathcal{G}$ is connected and $p^* \in \text{int}\Delta$, then p^* is asymptotically stable [4].

3. CONTROL PROBLEM STATEMENT

Consider a large-scale system composed by M controllable sub-systems that are connected by a communication network. The topology of the communication network is given by an undirected and connected graph denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $\mathcal{V} = \{1, ..., M\}$ be the set of nodes that represent the M sub-systems, and $\mathcal{E} \subset \{(i, j) : i, j \in \mathcal{V}\}$ the set of links representing the available communication and/or information sharing among sub-systems. Each controllable sub-system has a linear time-invariant discrete dynamics given by

$$x_i(k + 1) = A_i x_i(k) + B_i u_i(k), \tag{3}$$

where $k \in \mathbb{Z}_+$ denotes the discrete time, $i \in \mathcal{V}$ is the sub-system index, $x_i(k) \in \mathbb{R}^{n_i}$ denotes the state vector, $u_i(k) \in \mathbb{R}^{m_i}$ denotes the input vector of the i^{th} sub-system, and matrices $A_i \in \mathbb{R}^{n_i \times n_i}$ and $B_i \in \mathbb{R}^{n_i \times m_i}$ have constant elements. The optimization problem behind the

[1] Although the proposed methodology uses the distributed Smith dynamics, any distributed population dynamics from Barreiro-Gomez et al. (2014) can be implemented.
[2] Entities or elements capable to make decisions.
[3] Adapted from Theorem 3.1.3 and Corollary 3.1.4 in Sandholm (2010), where the corresponding proof can be found.
[4] The proof of this proposition is not presented due to the lack of space, and can be found in Barreiro-Gomez et al. (2014).

MPC controller can be stated with $m_i = 1$ for simplicity, and without loss of generality as follows:

$$\min_{\mathbf{u}} J(k) = \sum_{i=1}^{M} \left\{ \sum_{j=1}^{H_p} \|x_i(k+j) - r_i(k+j)\|_{Q_i}^2 + \sum_{j=0}^{H_p-1} \|u_i(k+j)\|_{R_i}^2 \right\}, \tag{4a}$$

subject to

$$x_i(k+1+j|k) = A_i x_i(k+j|k) + B_i u_i(k+j|k), \tag{4b}$$

$$x_i(k+j|k) \in \mathcal{X}_i, \tag{4c}$$

$$u_i(k+j|k) \in \mathcal{U}_i, \tag{4d}$$

$$\sum_{i=1}^{M} u_i(k+j|k) \leq K, \tag{4e}$$

where the constraints (4b) and (4c) for all $i \in \mathcal{V}$, $j \in [0, H_p] \subset \mathbb{Z}_+$; constraint (4d) for all $i \in \mathcal{V}$, $j \in [0, H_p - 1] \subset \mathbb{Z}_+$; and (4e) with $j \in [0, H_p - 1] \subset \mathbb{Z}_+$. The sets $\mathcal{X}_i \triangleq \{x_i \in \mathbb{R}^{n_i} : \underline{x}_i \leq x_i \leq \bar{x}_i\}$, and $\mathcal{U}_i \triangleq \{u_i \in \mathbb{R}^{m_i} : \underline{u}_i \leq u_i \leq \bar{u}_i\}$; and $\mathbf{u} \triangleq \{u(0|k), u(1|k), ..., u(H_p - 1|k)\}$ is an input sequence over a fixed-time prediction horizon H_p. Moreover, $Q_i \in \mathbb{R}^{n_i \times n_i}$ is a positive semi-definite weighting matrix related to the system states, and $R_i \in \mathbb{R}^{m_i \times m_i}$ is a positive definite weighting matrix related to the control actions. Vector $r_i(k)$ is the reference for the i^{th} sub-system. Vectors $\underline{x}_i$ and $\bar{x}_i$ determine the minimum and maximum state bounds of the i^{th} sub-system, respectively, and $\underline{u}_i$ and $\bar{u}_i$ determine the minimum and maximum control input bounds, respectively. The value of $K \in \mathbb{R}^m$ determines the total available resource as an energy constraint for the whole system. The cost function (4a) penalizes the state error and the energy of control actions for all sub-systems over H_p.

4. NON-CENTRALIZED MPC WITH POPULATION DYNAMICS

The topology for the non-centralized MPC controller with population dynamics implies a local MPC controller per sub–system. Each local controller needs to solve a partial problem and then coordinates its control signal by means of the DSD considering the constraint (4e). If the constraint (4e) is omitted, then the optimization problem (4) can be decoupled since sub-systems dynamics are decoupled as well as constraints (4b), (4c), and (4d). Consequently, a local MPC controller for the i^{th} sub-system can be designed with a cost function given by

$$\min_{\mathbf{u}} J_i(k) = \sum_{j=1}^{H_p} \|x_i(k+j) - r_i(k+j)\|_{Q_i}^2 + \sum_{j=0}^{H_p-1} \|u_i(k+j)\|_{R_i}^2, \tag{5}$$

subject to (4b), (4c), and (4d). In order to deal with the constraint (4e), a distributed full potential game with DSD is proposed. Since (4e) is not an equality constraint, it is necessary to add a slack variable denoted by p_{M+1} to the game, for which this slack variable is treated as a new node added to the graph and can be connected to any arbitrary node [5]. Additionally, its fitness function is chosen as $F_{M+1} = 0$. The slack variable allows to use less than the total available resource when convenient.

Notice that the solution of the MPC, denoted as u_i^*, is an input to the DSD, which computes in a distributed way the final optimal control action p_i^* applied to the associated sub–system. Furthermore, the local MPC controller supplies the bounds u_i^{min}, u_i^{max} for the corresponding control signal, such that the problem (5) is feasible. i.e., $u_i^{min}, u_i^{max} \in \mathcal{U}$, where $\mathcal{U}$ is the feasible set of control actions of the problem (5). The bound u_i^{min} is found with local information by solving the optimization problem (5) with weights $Q_i = \mathbf{0}_{n_i \times n_i}$ and $R_i = \mathbb{I}_{m_i \times m_i}$, where $\mathbf{0}_{n_i \times n_i}$ is a matrix with null entries, and $\mathbb{I}_{m_i \times m_i}$ is the identity matrix; and the bound u_i^{max} is found similarly with $Q_i = \mathbb{I}_{n_i \times n_i}$ and $R_i = \mathbf{0}_{m_i \times m_i}$. Both problems are solved subject to (4b), (4c), and (4d).

Remark 1. Generally, the bounds $\underline{u}_i$ and $\bar{u}_i$ in (4d) are different from bounds u_i^{min} and u_i^{max}. The values $\underline{u}_i$ and $\bar{u}_i$ determine the physical constraints for the control actions, whereas u_i^{min} and u_i^{max} determine the bounds of control actions that guarantee feasibility of (5). ♢

A strictly-concave full-potential function is proposed for the distributed population dynamics as follows:

$$f(p) = -\sum_{i=1}^{M} w_i (u_i^* - p_i)^2, \tag{6}$$

where w_i assigns a weighting factor to each control action, e.g., if $w_i = e_i$, for all $i \in S$, then more priority is assigned to those sub-systems with higher error. Consequently, the fitness functions for the game are given by $F(p) = \nabla f(p)$, i.e., $F_i(p_i) = 2w_i(u_i^* - p_i)$. Note that this methodology does not require full information of all control actions and/or all states of sub-systems since: *i)* the graph $\mathcal{G}$ representing the information interaction among sub-systems is a non-complete graph; and *ii)* the proposed fitness functions are decoupled, i.e., F_i depends only on information of the i^{th} sub-system. In order to satisfy the feasible region, the DSD are modified as follows:

$$\dot{p}_i = \sum_{j \in \mathcal{N}_i} (p_j - u_j^{min})[F_i - F_j]_+ - (p_i - u_i^{min}) \sum_{j \in \mathcal{N}_i} [F_j - F_i]_+. \tag{7}$$

Remark 2. It is known that $u_i^{min} \leq u_i^* \leq u_i^{max}$. There are two possible cases: *i)* when there is enough resource in the system, then $p_i^* = u_i^*$ and the final control action belongs to the feasible region; and *ii)* when there is not enough resource in the system, then $p_i^* < u_i^*$ and the term $p_i - u_i^{min}$ in (7) guarantees that the evolution of p_i will never be under the value of u_i^{min}, due to the fact that $\sum_{j \in \mathcal{N}_i} (p_j - u_j^{min})[F_i - F_j]_+ \geq 0$ in (7). In conclusion, the final control action p_i^* belongs to the feasible region. ♢

Each sub-system has a local MPC controller in which the optimization problem (5) is solved every $k \in \mathbb{Z}_+$, then there is a set of M controllers generating an optimal control action $u_i^*(k)$ for all $i \in S$. This optimal control action (with respect to (5)) provides a fitness function $F_i(p_i)$ to the DSD that calculate in a distributed way the final control action p_i^* satisfying the constraint (4e). Figure 1 shows the flow diagram of the proposed non-centralized MPC with DSD. The DSD require u_i^*, for all $i \in S$ to set the fitness functions in the game. The DSD also require the limits $[u_i^{min}, u_i^{max}]$ for all $i \in S$ in order to guarantee that the set of final control action p^* belongs to the feasible regions, i.e., $u_i^{min} \leq p_i^* \leq u_i^{max}$, for all $i \in S$. Moreover, the constraint satisfaction problem (CSP) must be solved in a distributed way since there is not full information in the non-centralized configuration.

[5] Since this slack node is added in the design stage, it is desired that the connectivity of the graph does not depend on this slack node.

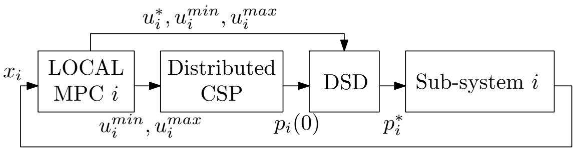

Fig. 1. Flow diagram of the proposed methodology.

4.1 Plug-and-play Property

One of the advantages of the decentralized control design with population dynamics is the reduction of information dependence. Furthermore, there is another relevant advantage associated to the proposed scheme. The methodology consists in dividing the original problem into different sub–problems whose solutions are coordinated to obtain a final control action. In this regard, each control problem associated to each sub–system is independent from others.

Now, suppose that a new sub–system is added to the initial problem (4), i.e., that the number of sub–systems is $M+1$, and then only sums involved in (4a) and (4b) should be modified in the MPC optimization problem. Notice that for this new optimization problem, the decoupled set of optimization problems is the same as in (5), including the optimization problem associated to the sub–system $M+1$, and the bounds u_{M+1}^{min}, and u_{M+1}^{max} can be found without requiring information from the other sub–systems. Finally, a new node is added to the graph $\mathcal{G}$ and the CSP computes a feasible initial condition for the DSD. Consequently, the proposed control scheme is plug and play since it is not necessary to modify previously already designed parts of the MPC controller in order to add a new sub–system to the problem. The same analysis may be done for the removal of sub–systems to the problem, but it has to be taken into account that the graph $\mathcal{G}$ cannot be disconnected in this modification.

4.2 Control Convergence Cases

Once the methodology has been presented, this subsection is dedicated to analyze two possible cases that might occur when computing the optimal control action in a distributed way with the proposed methodology.

Case 1: $u_i^* > u_i^{min}$, for all $i \in \mathcal{V}$. It is guaranteed that the optimal point $p^* \in \text{int}\Delta$, the graph $\mathcal{G}$ is connected for all t since u^* is an interior point of $\mathcal{U}$, and Proposition 2 holds.
Case 2: $u_i^* = u_i^{min}$, for any $i \in \mathcal{V}$. The optimal control action u^* is at the limit of $\mathcal{U}$, i.e., there is an active constraint. Consequently, the node associated to that decision variable disappears and $\mathcal{G}$ might get disconnected depending on its topology. Then, each problem in each sub–complete graph $\tilde{\mathcal{G}} \subset \mathcal{G}$ converges to an optimal solution, and the global problem is solved with sub–optimal solution. However, an appropriate design of redundant links might solve this inconvenience.

4.3 Constraint Satisfaction Problem (CSP)

As it was stated in Proposition 2, an important feature of the DSD given in (7) is that they guarantee constraint satisfaction along the time provided that the initial condition belongs to the feasible region of the considered problem. Therefore, an initial feasible point should be found in order to initialize the DSD, i.e., $p(0)$ must satisfy the following constraints:

$$\begin{aligned} u_i^{min} \leq p_i(0) \leq u_i^{max} \quad &\text{(feasibility of the local MPC controller)} \\ p_{M+1}(0) \geq 0 \quad &\text{(positivity of the slack variable)} \\ \sum_{i=1}^{M+1} p_i(0) = K. \quad &\text{(resource constraint)} \end{aligned} \tag{8}$$

The above requirements are not trivial since a distributed framework in which each node of the network only has partial information of the whole problem is considered. This section describes a method (inspired by the algorithm proposed in (Garin and Schenato (2010))) that solves the CSP characterized by the constraints (8) in a distributed way. This methodology is used before applying the DSD.

First of all, consider the information that each node has: *i)* the i^{th} node knows its local bounds, i.e., u_i^{min} and u_i^{max}; *ii)* it is assumed that the slack node knows the available resource K; and *iii)* the nodes can share information by using the communication network that is given by the connected graph $\mathcal{G}$. Assuming that there exists a vector $p(0)$ that satisfies the constraints in (8), a possible choice for $p(0)$ is as follows:

$$\begin{aligned} p_i(0) &= u_i^{min}, \text{ for all } i = 1, \ldots, M, \\ p_{M+1}(0) &= K - \xi, \end{aligned} \tag{9}$$

where $\xi = \sum_{i=1}^{M} u_i^{min}$. Notice that this solution can be computed directly by using only local information except for the term ξ. Therefore, the idea is that the $(M+1)^{th}$ node obtains ξ by means of a distributed algorithm. In order to do so, ξ is rewritten as the product of the total number of nodes by the average of the minimum boundaries of nodes, i.e.,

$$\xi = \sum_{i=1}^{M} u_i^{min} = (M+1)\text{mean}(u_1^{min}, \ldots, u_M^{min}, 0), \tag{10}$$

where mean$(\cdot)$ denotes the arithmetic mean. Now, the original problem is divided into two sub-problems: *i)* find the average of the lower bounds of nodes; and *ii)* find the total number of nodes. Notice that each problem needs a distributed solution.

Finding the average of the lower bounds of nodes: The main idea is that the information about the lower bounds of nodes propagates through the network. For this purpose, an auxiliary variable $\xi_i^{min} \in \mathbb{R}$ per node is defined, where the subindex i denotes that the variable is associated with the i^{th} node. These variables are initialized with the corresponding node lower bound as follows: $\xi_i^{min}(0) = u_i^{min}$, for all $i = 1, \ldots, M$, and $\xi_{M+1}^{min}(0) = 0$. Notice that the arithmetic mean of the lower bounds of nodes is equal to the arithmetic mean of the initial conditions of the auxiliary variables defined above, i.e., mean$(u_1^{min}, \ldots, u_M^{min}, 0)$ = mean$(\xi_1^{min}(0), \ldots, \xi_{M+1}^{min}(0))$. In order to calculate this quantity in a distributed way, a standard average consensus algorithm as

$$\dot{\xi}_i^{min} = \sum_{j \in \mathcal{N}_i} \left(\xi_j^{min} - \xi_i^{min} \right) \tag{11}$$

can be applied (Olfati-Saber et al. (2007)). According to Olfati-Saber et al. (2007), if the communication network is described by a connected graph, then $\xi_i^{min^*}$ = mean$(\xi_1^{min}(0), \ldots, \xi_{M+1}^{min}(0))$, for all $i = 1, \ldots, M+1$, where ξ_i^{min*} is the steady state value of ξ_i^{min}. Thus, $\xi_i^{min*} = \text{mean}(u_1^{min}, \dots, u_M^{min}, 0)$. This implies that the $(M+1)^{th}$ node is capable to obtain the required value by using only local information.

Finding the total number of nodes: The second problem is to locally compute the total number of nodes. In order to do so, a similar procedure as in the previous problem is followed. Define another auxiliary variable per node. Let ξ_i^c be the variable associated with the i^{th} node. These new auxiliary variables are initialized as follows: $\xi_i^c(0) = 0$, for all $i = 1, \dots, M$, and $\xi_{M+1}^c(0) = 1$. The above initialization values are important since their average is related to the required information, i.e., $\text{mean}(\xi_1^c(0), \dots, \xi_{M+1}^c(0)) = (M+1)^{-1}$. Thus, the same algorithm as in (11) can be applied to compute the needed quantity in a distributed way, i.e.,

$$\dot{\xi}_i^c = \sum_{j \in \mathcal{N}_i} \left(\xi_j^c - \xi_i^c\right). \quad (12)$$

Again, $\xi_i^{c*} = \text{mean}(\xi_1^c(0), \dots, \xi_{M+1}^c(0)) = (M+1)^{-1}$, for all $i = 1, \dots, M+1$, in steady state. Therefore, the $(M+1)^{th}$ node can obtain the required quantity by taking $({\xi_{M+1}^c}^*)^{-1}$. Summarizing, the steady state solutions of (11)–(12) are used to compute ξ in (10), i.e., $\xi = \xi_i^{min*}/\xi_i^{c*}$, and then this result is replaced in (9) in order to obtain the required initial feasible point for the DSD. Notice that this procedure is done by using only local information.

5. CASE STUDY

The proposed case study is an industrial process that comprises four continuously stirred tank reactors (CSTR) as shown in Fig. 2. This is a proof–of–concept problem to illustrate the proposed methodology performance. Moreover, the methodology is scalable to any higher dimension. The control objective is to maintain the concentrations C_{A1}, C_{A2}, C_{A3}, and C_{A4} as close as possible to the references $0.1, 0.15, 0.2$, and 0.25, respectively.

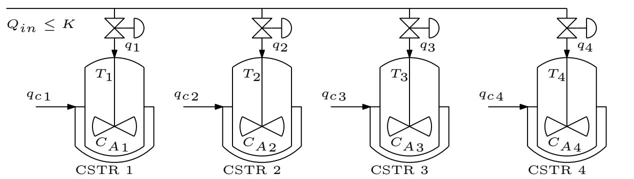

Fig. 2. Four CSTR controlling concentration C_{Ai} with q_{ci}, $i = 1, \dots, 4$, and inflow resource $Q_{in} = \sum_{i=1}^4 q_i$.

Additionally, the system has a limited total inflow resource for the control inputs q_1, q_2, q_3, and q_4 given by $K = 750$ l/min. The physical constraints for the inflows are given by the range $[\underline{u}_i, \bar{u}_i] = [0, 300]$ (in l/min).The discrete-time linear model of the form (3) with a sampling time $T_s = 0.1$min, for each CSTR around the operational point given by $\tilde{C}_{Ai} = 0.0823$ mol/l, $\tilde{T}_i = 442$ K, and $\tilde{q}_{ci} = 100$ mol/l is given by the following matrices:

$$A_i = \begin{bmatrix} 0.33 & 1.29\text{e}{-5} \\ 0.61 & 2.45\text{e}{-5} \end{bmatrix}, \; B_i = \begin{bmatrix} 5.49\text{e}{-4} \\ 1.95\text{e}{-4} \end{bmatrix},$$

where the output is $y_i(k) = x_i(k) = [C_{Ai} \;\; T_i]^\top$, for all $i = 1, ..., 4$, and $u_i(k) = q_i(k)$. In order to analyze the performance of the proposed non–centralized MPC approach, three different scenarios are considered, i.e.,

- *Scenario 1:* Centralized MPC without the resource constraint (4e), and $H_p = 3$.
- *Scenario 2:* Centralized MPC with the resource constraint (4e), and $H_p = 3$.
- *Scenario 3:* Non-centralized MPC with DSD and the resource constraint (4e), and $H_p = 3$.

The selected communication topology is given by the non-complete graph with $\mathcal{V} = \{1, 2, 3, 4\}$ and $\mathcal{E} = \{(1, 2), (2, 3), (3, 4)\}$ (path graph). The weighting factors in (6) are chosen as $w_i = e_i$, where e_i is the error of the i^{th} sub-system. This is made to assign the limited available resource according to how different states are from the desired references, i.e., higher priority is assigned to sub-systems with more error. Also, notice that the weight w_i varies along the time according to the dynamical behavior of the system, i.e., the potential function has a dynamical tuning.

Results and Discussion

Figure 3 shows the evolution of concentration at the four CSTR, and the applied q_1, q_2, q_3, and q_4 for the three scenarios. As it is expected, the concentration of each CSTR reaches its corresponding reference when the total inflow is unconstrained. If this is not the case (i.e., when the inflows are limited to a value lower than 750 l/min), concentrations are below their corresponding set–points. This is because there is not enough feed flow rate in the reactor mass balance to make the concentration increase to the desired value. However, in the latter situation, controllers tries to use all the available resource to keep the controlled variables close to the desired state.

Table 1. Steady state error for the three MPC schemes.

CSTR	Scenario 1	Scenario 2	Scenario 3
1	0	0.0145	0.0222
2	0	0.0192	0.0228
3	0	0.0243	0.0218
4	0	0.0322	0.0233
Sum	0	0.0902	0.0901

Table 1 shows the error in steady state for the three MPC controllers. As it was previously mentioned, the MPC without the resource constraint achieves the reference. On the other hand, the error when there is a resource coupled constraint is not null. Furthermore, note that the accumulative error is the same for both the centralized MPC with coupled constraint and the non-centralized MPC with schemes DSD and coupled constraint. However, the errors for the non-centralized MPC with DSD are equitable throughout the sub-systems since the weights at the potential function (6) are chosen as $w_i = e_i$. This fact is appealing since it shows a fair distribution of the available resource (i.e., there is no preferences for one sub–system over the others), and this is obtained without modifying the local MPC controllers tuning. Figure 4 shows the total resource used to control the four CSTR. It can be seen that to achieve all references, it is necessary to have a total resource of $K = 861$ l/min. With the centralized MPC controller, the constraint of resource limited to $K \leq 750$ l/min is satisfied.

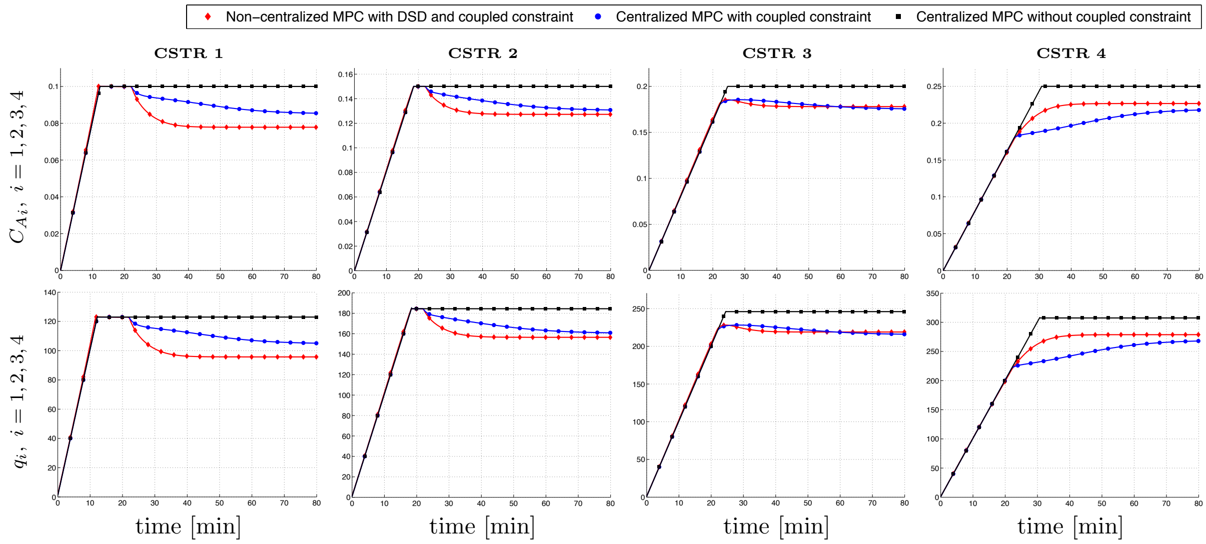

Fig. 3. Four CSTR. Control of concentration C_{Ai} with q_i, $i = 1, \ldots, 4$.

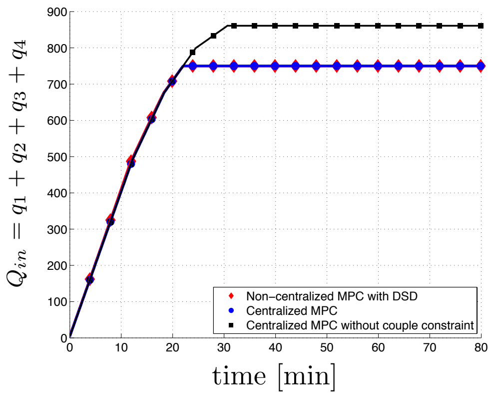

Fig. 4. Total resource for the MPC schemes.

6. CONCLUSIONS

This work has proposed a novel methodology based on distributed population dynamics to make non-centralized an MPC scheme with a single coupled constraint associated to a limited resource. Results have shown that the methodology satisfies the coupled constraint in a distributed way. Simulations have shown that choosing the error as a weighting parameter, the same error is obtained in steady state for all sub-systems. The DSD have been selected to design the non-centralized MPC controller. However, the same technique can be extended to other distributed population dynamics Results have also shown that the settling time for the non-centralized MPC scheme with DSD and coupled constraint is shorter than the centralized MPC with coupled constraint, i.e., states achieve the stationary condition faster. This is because a higher inflow is applied to the system at the beginning.

REFERENCES

Barreiro-Gomez, J., Obando, G., and Quijano, N. (2014). Distributed population dynamics: Optimization and control applications. *IEEE Transactions on Systems, Man, and Cybernetics (Submitted)*.

Christofides, P.D., Scattolini, R., Muñoz de la Peña, D., and Liu, J. (2013). Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51, 21–41.

Conte, C., Summers, T., Zeilinger, M., Morari, M., and Jones, C. (2012). Computational Aspects of Distributed Optimization in Model Predictive Control. In *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)*, 6819–6824.

Garin, F. and Schenato, L. (2010). A survey on distributed estimation and control applications using linear consensus algorithms. *Networked Control Systems*, 406, 75–107.

Marden, J. and Shamma, J. (2015). Game theory and distributed control. In *Handbook of Game Theory*, volume 4, 861–899. Elsevier Science.

Obando, G., Pantoja, A., and Quijano, N. (2014). Building Temperature Control based on Population Dynamics. *IEEE Transactions on Control Systems Technology.*, 22(1), 404–412.

Ocampo-Martinez, C., Bovo, S., and Puig, V. (2011). Partitioning approach oriented to the decentralised predictive control of large-scale systems. *Journal of Process Control*, 21(5), 775–786.

Olfati-Saber, R., Fax, J., and Murray, R. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215–233.

Pantoja, A. and Quijano, N. (2011). A population dynamics approach for the dispatch of distributed generators. *IEEE Transactions on Industrial Electronics*, 58(10), 4559–4567.

Sandholm, W.H. (2010). *Population games and evolutionary dynamics*. Economic learning and social evolution. Cambridge, Mass. MIT Press.

Spudic, V. and Baotić, M. (2013). Fast coordinated model predictive control of large-scale distributed systems with single coupling constraint. In *Proceedings of the European Control Conference (ECC)*, 2783–2788.

Tembine, H., Altman, E., El-Azouzi, R., and Hayel, Y. (2010). Evolutionary games in wireless networks. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*, 40(3), 634–646.

Wakasa, Y., Arakawa, M., Tanaka, K., and Akashi, T. (2008). Decentralized model predictive control via dual decomposition. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC)*, 381–386.